***EJERCICIOS PARA RESOLVER / ESTRUCTURAS DE CONTROL***

**Ejercicio 3.13**.- El DNI consta de un número entero de 8 dígitos seguido de una letra que se obtiene a partir del número utilizando la siguiente fórmula: numero\_letra = numero\_DNI módulo 23 El número de la letra se obtiene a partir de la tabla: Donde la letra se asigna a partir del valor obtenido para numero\_letra. Por ejemplo: Para el DNI 12.345.678 tendríamos: numero\_letra = 12345678 % 23 = 14 Al 14 le corresponde en la tabla la letra Z. La salida sería: NIF: 12345678 - Z 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 T R W A G M Y F P D X B N J Z

public static char calcularLetraDNI(int dni) {

String letras = "TRWAGMYFPDXBNJZSQVHLCKE";

int indice = dni % 23;

char letra = letras.charAt(indice);

return letra;

}

int dni = 12345678;

char letra = calcularLetraDNI(dni);

System.out.println("La letra del DNI " + dni + " es " + letra);

**Ejercicio 3.14**.- Pedir un número entre 0 y 99 y mostrarlo escrito. Por ejemplo, para 56 mostrar: cincuenta y seis. (Para simplificar mostrar los números de la decena como diez y uno, diez y dos, … en lugar de once, doce,…)

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

String[] units = {"cero", "uno", "dos", "tres", "cuatro", "cinco", "seis", "siete", "ocho", "nueve"};

String[] tens = {"diez", "once", "doce", "trece", "catorce", "quince", "dieciséis", "diecisiete", "dieciocho", "diecinueve"};

String[] multiplesOfTen = {"", "", "veinte", "treinta", "cuarenta", "cincuenta", "sesenta", "setenta", "ochenta", "noventa"};

System.out.print("Ingresa un número entre 0 y 99: ");

int num = input.nextInt();

if (num < 10) {

System.out.println(units[num]);

} else if (num < 20) {

System.out.println(tens[num - 10]);

} else if (num % 10 == 0) {

System.out.println(multiplesOfTen[num / 10]);

} else {

System.out.println(multiplesOfTen[num / 10] + " y " + units[num % 10]);

}

}

}

**Ejercicio 3.15**.- Utiliza el operador ternario para calcular el valor absoluto de un número que se solicita al usuario por teclado

int num = -10;

int absNum = Math.abs(num);

System.out.println("El valor absoluto de " + num + " es " + absNum);

otra forma:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Ingresa un número: ");

int num = input.nextInt();

int absValue = num >= 0 ? num : -num;

System.out.println("El valor absoluto de " + num + " es " + absValue);

}

}

**Ejercicio 3.16**.- Realiza el «juego de la suma», que consiste en que aparezcan dos números aleatorios comprendidos entre 1 y 99 que el usuario tiene que sumar. La aplicación debe indicar si el resultado de la operación es correcto o incorrecto

import java.util.Random;

import java.util.Scanner;

public class JuegoDeLaSuma {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

Random rnd = new Random();

// Generar dos números aleatorios entre 1 y 99

int num1 = rnd.nextInt(99) + 1;

int num2 = rnd.nextInt(99) + 1;

// Mostrar la operación al usuario y pedir la respuesta

System.out.printf("¿Cuánto es %d + %d? ", num1, num2);

int respuesta = sc.nextInt();

// Comprobar si la respuesta es correcta

int suma = num1 + num2;

if (respuesta == suma) {

System.out.println("¡Respuesta correcta!");

} else {

System.out.println("Respuesta incorrecta.");

System.out.printf("La respuesta correcta es %d + %d = %d", num1, num2, suma);

}

sc.close();

}

}

**Ejercicio 3.17**.- Modifica el código del ejercicio 3.16 para que, además de los dos números aleatorios, también aparezca la operación que debe realizar el jugador: suma, resta o multiplicación.

import java.util.Random;

import java.util.Scanner;

public class JuegoDeLaOperacion {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

Random rnd = new Random();

// Generar dos números aleatorios entre 1 y 99

int num1 = rnd.nextInt(99) + 1;

int num2 = rnd.nextInt(99) + 1;

// Generar un número aleatorio entre 1 y 3 para seleccionar la operación

int operacion = rnd.nextInt(3) + 1;

String simbolo;

int resultado;

switch (operacion) {

case 1: // Suma

simbolo = "+";

resultado = num1 + num2;

break;

case 2: // Resta

simbolo = "-";

resultado = num1 - num2;

break;

default: // Multiplicación

simbolo = "\*";

resultado = num1 \* num2;

break;

}

// Mostrar la operación al usuario y pedir la respuesta

System.out.printf("¿Cuánto es %d %s %d? ", num1, simbolo, num2);

int respuesta = sc.nextInt();

// Comprobar si la respuesta es correcta

if (respuesta == resultado) {

System.out.println("¡Respuesta correcta!");

} else {

System.out.println("Respuesta incorrecta.");

System.out.printf("La respuesta correcta es %d %s %d = %d", num1, simbolo, num2, resultado);

}

sc.close();

}

}

**Ejercicio 3.18.**- Crea una aplicación que solicite al usuario cuántos grados tiene un ángulo y muestre el resultado equivalente en radianes. Si el ángulo introducido por el usuario no se encuentra en el rango entre 0º y 360º, hay que transformarlo a dicho rango. En este ejercicio puede ser útil el operador módulo.

import java.util.Scanner;

public class AnguloEnRadianes {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Pedir al usuario el ángulo en grados

System.out.print("Introduce el ángulo en grados: ");

double grados = sc.nextDouble();

// Asegurarse de que el ángulo está en el rango de 0º a 360º

grados = grados % 360;

if (grados < 0) {

grados += 360;

}

// Convertir el ángulo a radianes

double radianes = Math.toRadians(grados);

// Mostrar el resultado

System.out.printf("%.2f grados son equivalentes a %.2f radianes.", grados, radianes);

sc.close();

}

}

**Ejercicio 3.33.**- Realiza un programa que convierta un número decimal en su representación binaria. Hay que tener en cuenta que desconocemos cuántas cifras tiene el número que introduce el usuario. Por simplicidad mostraremos el número binario con un dígito por línea.

import java.util.Scanner;

public class DecimalABinario {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Pedir al usuario el número decimal

System.out.print("Introduce un número decimal: ");

int decimal = sc.nextInt();

// Convertir el número decimal a binario

String binario = "";

while (decimal > 0) {

int resto = decimal % 2;

binario = resto + "\n" + binario;

decimal = decimal / 2;

}

// Mostrar el resultado

System.out.printf("El número %d en binario es:\n%s", decimal, binario);

sc.close();

}

}

**Ejercicio 3.34.**- Modifica el código del Ejercicio 3.33 para que el usuario pueda introducir un número en binario y el programa muestre su conversión a decimal.

import java.util.Scanner;

public class BinarioADecimal {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Pedir al usuario el número en binario

System.out.print("Introduce un número en binario: ");

String binario = sc.nextLine();

// Convertir el número binario a decimal

int decimal = 0;

int potencia = 0;

for (int i = binario.length() - 1; i >= 0; i--) {

int digito = Character.getNumericValue(binario.charAt(i));

decimal += digito \* Math.pow(2, potencia);

potencia++;

}

// Mostrar el resultado

System.out.printf("El número %s en decimal es: %d", binario, decimal);

sc.close();

}

}

**Ejercicio 3.35.**- Escribe un programa que incremente la hora de un reloj. Se pedirán por teclado la hora, minutos y segundos, así como cuántos segundos se desea incrementar la hora introducida. La aplicación mostrará la nueva hora. Por ejemplo, si las 13:59:51 se incrementan en 10 segundos, resultan las 14:00:01.

import java.util.Scanner;

public class IncrementarHora {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int horas, minutos, segundos, segundosIncremento;

System.out.print("Introduzca la hora: ");

horas = sc.nextInt();

System.out.print("Introduzca los minutos: ");

minutos = sc.nextInt();

System.out.print("Introduzca los segundos: ");

segundos = sc.nextInt();

System.out.print("Introduzca la cantidad de segundos que desea incrementar: ");

segundosIncremento = sc.nextInt();

// Convertir todo a segundos

int totalSegundos = horas\*3600 + minutos\*60 + segundos + segundosIncremento;

// Calcular la nueva hora

int nuevaHora = (totalSegundos / 3600) % 24;

int nuevosMinutos = (totalSegundos % 3600) / 60;

int nuevosSegundos = totalSegundos % 60;

// Mostrar la nueva hora

System.out.println("La nueva hora es: " + nuevaHora + ":" + nuevosMinutos + ":" + nuevosSegundos);

}

}

**Ejercicio 3.36.**- Realiza un programa que nos pida un número n y nos diga cuántos números hay entre 1 y n que sean primos. Un número primo es aquel que solo es divisible por 1 y por sí mismo. Por ejemplo, con n = 8 la salida sería:

import java.util.Scanner;

public class ContarPrimos {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n, contador = 0;

System.out.print("Introduzca un número: ");

n = sc.nextInt();

for (int i = 1; i <= n; i++) {

if (esPrimo(i)) {

contador++; } }

System.out.println("Hay " + contador + " números primos entre 1 y " + n);

}

public static boolean esPrimo(int numero) {

if (numero <= 1) {

return false; }

for (int i = 2; i <= Math.sqrt(numero); i++) {

if (numero % i == 0) { return false;

}

} return true;

}

**Ejercicio 3.37**.- Diseña una aplicación que dibuje el triángulo de Pascal para n filas. Numerando las filas y elementos desde 0, la fórmula para obtener el m-ésimo elemento de la n-ésima fila es: , = ! ! − ! Siendo n! el factorial de n.

import java.util.Scanner;

public class TrianguloPascal {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n;

System.out.print("Introduzca el número de filas: ");

n = sc.nextInt();

for (int i = 0; i < n; i++) {

for (int j = 0; j <= i; j++) {

System.out.print(calcularElemento(i, j) + " ");

}

System.out.println();

}

}

public static int calcularElemento(int n, int m) {

return factorial(n) / (factorial(m) \* factorial(n - m));

}

public static int factorial(int n) {

int resultado = 1;

for (int i = 2; i <= n; i++) {

resultado \*= i;

}

return resultado;

}

}

**Ejercicio 3.38.**- Solicita al usuario un número n y dibuja un triángulo de base y altura n. Por ejemplo, para n = 4:

import java.util.Scanner;

public class Triangulo {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n;

System.out.print("Introduzca el número de filas: ");

n = sc.nextInt();

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= n - i; j++) {

System.out.print(" ");

}

for (int j = 1; j <= 2 \* i - 1; j++) {

System.out.print("\*");

}

System.out.println();

}

}

}

**Ejercicio 3.39**.- Para dos números dados a y b, es posible buscar el máximo común divisor (el número más grande que divide a ambos) mediante un algoritmo ineficiente pero sencillo: desde el menor de a y b, ir buscando, de forma decreciente, el primer número que divide a ambos simultáneamente. Realiza un programa que calcule el máximo común divisor de dos números.

import java.util.Scanner;

public class MaximoComunDivisor {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int a, b;

System.out.print("Introduzca el primer número: ");

a = sc.nextInt();

System.out.print("Introduzca el segundo número: ");

b = sc.nextInt();

int minimo = Math.min(a, b);

int mcd = 0;

for (int i = minimo; i >= 1; i--) {

if (a % i == 0 && b % i == 0) {

mcd = i;

break;

}

}

System.out.println("El máximo común divisor de " + a + " y " + b + " es " + mcd);

}

}

**Ejercicio 3.40**.- De forma análoga al Ejercicio 3.39 implementa un algoritmo que calcule el mínimo común múltiplo de dos números dados.

import java.util.Scanner;

public class MinimoComunMultiplo {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int a, b;

System.out.print("Introduzca el primer número: ");

a = sc.nextInt();

System.out.print("Introduzca el segundo número: ");

b = sc.nextInt();

int maximo = Math.max(a, b);

int mcm = maximo;

while (mcm % a != 0 || mcm % b != 0) {

mcm += maximo;

}

System.out.println("El mínimo común múltiplo de " + a + " y " + b + " es " + mcm);

}

}

**Ejercicio 3.41.**- Calcula la raíz cuadrada de un número natural mediante aproximaciones. En el caso de que no sea exacta, muestra el resto. Por ejemplo, para calcular la raíz cuadrada de 23 probamos 12 = 1, 22 = 4, 32 = 9, 42 = 16, 52 = 25 (nos pasamos), resultando 4 la raíz cuadrada de 23 con un resto de (23 – 16 = 7).

import java.util.Scanner;

public class RaizCuadrada {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n;

System.out.print("Introduzca un número natural: ");

n = sc.nextInt();

double raiz = calcularRaiz(n);

double resto = n - raiz \* raiz;

System.out.println("La raíz cuadrada de " + n + " es " + raiz);

if (resto != 0) {

System.out.println("El resto es " + resto);

}

}

public static double calcularRaiz(int n) {

double x = n;

double y = 1;

double error = 0.0001;

while (x - y > error) { x = (x + y) / 2; y = n / x; }

return x; } }

**Ejercicio 3.42**.- Escribe un programa que solicite al usuario las distintas cantidades de dinero de las que dispone. Por ejemplo: la cantidad de dinero que tiene en el banco, en una hucha, en un cajón y en los bolsillos. La aplicación mostrará la suma total de dinero de la que dispone el usuario. La manera de especificar que no desea introducir más cantidades es mediante el cero.

import java.util.Scanner;

public class SumaDinero {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

double cantidad, total = 0;

do {

System.out.print("Introduzca una cantidad de dinero (0 para terminar): ");

cantidad = sc.nextDouble();

total += cantidad;

} while (cantidad != 0);

System.out.println("La suma total de dinero es: " + total);

}

}

**Ejercicio 3.54.**- Diseña una función que calcule y muestre la superficie y el volumen de una esfera. Escribe un programa que solicite al usuario el radio y muestre los resultados usando la función.

import java.util.Scanner;

public class Esfera {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

double radio;

System.out.print("Introduzca el radio de la esfera: ");

radio = sc.nextDouble();

mostrarResultados(radio);

}

public static void mostrarResultados(double radio) {

double superficie = 4 \* Math.PI \* radio \* radio;

double volumen = 4.0 / 3.0 \* Math.PI \* radio \* radio \* radio;

System.out.println("Superficie: " + superficie);

System.out.println("Volumen: " + volumen);

}

}

**Ejercicio 3.55.**- Implementa la función static double distancia(double x1, double y1, double x2, double y2) que calcule y devuelva la distancia euclídea entre dos puntos (x1, y1) y (x2, y2). Escribe un programa que solicite al usuario las coordenadas de los dos puntos y devuelva la distancia entre ellos usando esta función.

import java.util.Scanner;

public class DistanciaEntrePuntos {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

double x1, y1, x2, y2;

System.out.print("Introduzca las coordenadas del primer punto (x1 y y1): ");

x1 = sc.nextDouble();

y1 = sc.nextDouble();

System.out.print("Introduzca las coordenadas del segundo punto (x2 y y2): ");

x2 = sc.nextDouble();

y2 = sc.nextDouble();

double distancia = distancia(x1, y1, x2, y2);

System.out.println("La distancia entre los puntos es: " + distancia);

}

public static double distancia(double x1, double y1, double x2, double y2) {

double dx = x2 - x1;

double dy = y2 - y1;

return Math.sqrt(dx \* dx + dy \* dy);

}

}

**Ejercicio 3.56.**- Escribe una función a la que se le pase como parámetros de entrada una cantidad de días, horas y minutos. La función calculará y devolverá el número de segundos que hay entre los datos de entrada. Implementa un programa que solicite al usuario los datos por teclado y devuelva los resultados usando la función.

import java.util.Scanner;

public class ConversionTiempo {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int dias, horas, minutos;

System.out.print("Introduzca la cantidad de días: ");

dias = sc.nextInt();

System.out.print("Introduzca la cantidad de horas: ");

horas = sc.nextInt();

System.out.print("Introduzca la cantidad de minutos: ");

minutos = sc.nextInt();

int segundos = calcularSegundos(dias, horas, minutos);

System.out.println("La cantidad de segundos es: " + segundos);

}

public static int calcularSegundos(int dias, int horas, int minutos) {

int totalSegundos = dias \* 24 \* 60 \* 60 + horas \* 60 \* 60 + minutos \* 60;

return totalSegundos;

}

}

**Ejercicio 3.57.**- Diseña una función a la que se le pasan las horas y minutos de dos instantes de tiempo con el siguiente prototipo: static int diferenciaMin(int hora1, int minuto1, int hora2, int minuto2) La función devolverá la cantidad de minutos que existen de diferencia entre los dos instantes. Escribe un programa que solicite al usuario los datos de los dos instantes y devuelva el resultado usando la función.

import java.util.Scanner;

public class DiferenciaTiempo {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int hora1, minuto1, hora2, minuto2;

System.out.print("Introduzca la hora del primer instante: ");

hora1 = sc.nextInt();

System.out.print("Introduzca los minutos del primer instante: ");

minuto1 = sc.nextInt();

System.out.print("Introduzca la hora del segundo instante: ");

hora2 = sc.nextInt();

System.out.print("Introduzca los minutos del segundo instante: ");

minuto2 = sc.nextInt();

int diferencia = diferenciaMin(hora1, minuto1, hora2, minuto2);

System.out.println("La diferencia de tiempo es de " + diferencia + " minutos.");

}

public static int diferenciaMin(int hora1, int minuto1, int hora2, int minuto2) {

int totalMinutos1 = hora1 \* 60 + minuto1;

int totalMinutos2 = hora2 \* 60 + minuto2;

int diferencia = Math.abs(totalMinutos1 - totalMinutos2);

return diferencia;

}

}

**Ejercicio 3.58**.- Implementa la función divisoresPrimos() que muestra, por consola, todos los divisores primos del número que se le pasa como parámetro. Escribe un programa que solicite al usuario el número y devuelva todos sus divisores primos usando la función

import java.util.Scanner;

public class DivisoresPrimos {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int numero;

System.out.print("Introduzca un número entero positivo: ");

numero = sc.nextInt();

System.out.print("Los divisores primos de " + numero + " son: ");

divisoresPrimos(numero);

}

public static void divisoresPrimos(int numero) {

for (int i = 2; i <= numero; i++) {

if (numero % i == 0) {

boolean esPrimo = true;

for (int j = 2; j < i; j++) {

if (i % j == 0) {

esPrimo = false;

break;

}

}

if (esPrimo) {

System.out.print(i + " ");

}

}

}

}

}

**Ejercicio 3.59.**- Escribe una función que muestre por consola una serie de números aleatorios enteros. Los parámetros de la función serán: la cantidad de números aleatorios que se mostrarán y los valores mínimos y máximos que estos pueden tomar. Escribe un programa que solicite al usuario los datos y escriba los números aleatorios usando la función.

import java.util.Scanner;

public class NumerosAleatorios {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int cantidad, minimo, maximo;

System.out.print("Introduzca la cantidad de números aleatorios a mostrar: ");

cantidad = sc.nextInt();

System.out.print("Introduzca el valor mínimo que pueden tomar los números: ");

minimo = sc.nextInt();

System.out.print("Introduzca el valor máximo que pueden tomar los números: ");

maximo = sc.nextInt();

numerosAleatorios(cantidad, minimo, maximo);

}

public static void numerosAleatorios(int cantidad, int minimo, int maximo) {

for (int i = 0; i < cantidad; i++) {

int numero = (int) (Math.random() \* (maximo - minimo + 1) + minimo);

System.out.print(numero + " ");

}

}

}

**Ejercicio 3.60.**- Sobrecarga la función del ejercicio anterior para que el único parámetro sea la cantidad de números aleatorios que se muestran por consola. Los números aleatorios serán reales y estarán comprendidos entre 0 y 1. Escribe un programa que sea capaz de utilizar las dos funciones sobrecargadas

import java.util.Scanner;

public class NumerosAleatorios {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int cantidad;

System.out.print("Introduzca la cantidad de números aleatorios a mostrar: ");

cantidad = sc.nextInt();

numerosAleatorios(cantidad);

System.out.println();

numerosAleatorios(cantidad, 0, 1);

}

public static void numerosAleatorios(int cantidad) {

for (int i = 0; i < cantidad; i++) {

double numero = Math.random();

System.out.print(numero + " ");

}

}

public static void numerosAleatorios(int cantidad, double minimo, double maximo) {

for (int i = 0; i < cantidad; i++) {

double numero = Math.random() \* (maximo - minimo) + minimo;

System.out.print(numero + " ");

}

}

}